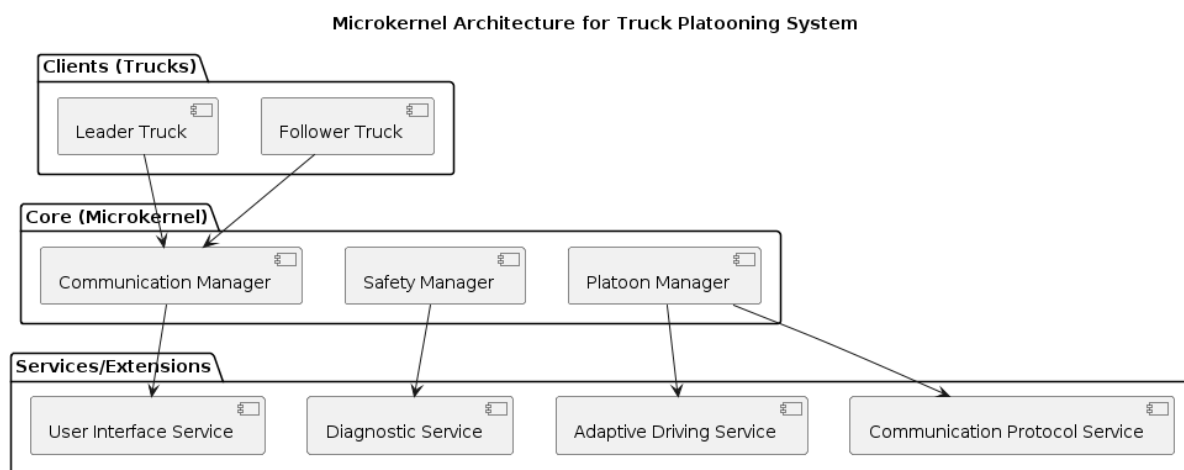


[Only for reference purpose]

Designing a Truck Platooning Systems

Message Queuing Pattern (from Concurrency Architectural design pattern) used in Microkernel pattern configuration.

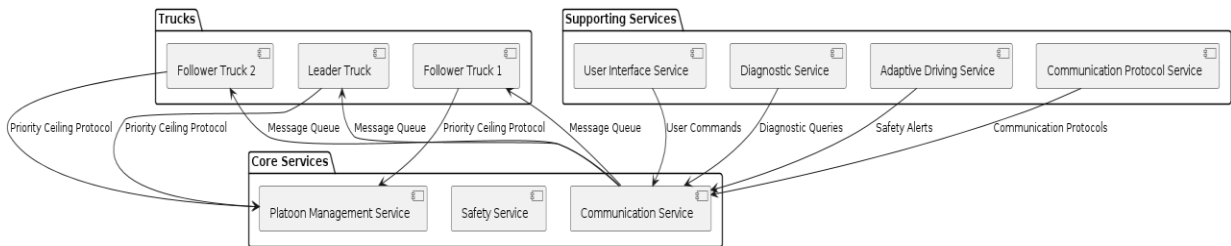
Microkernel Architecture Pattern for Truck Platooning



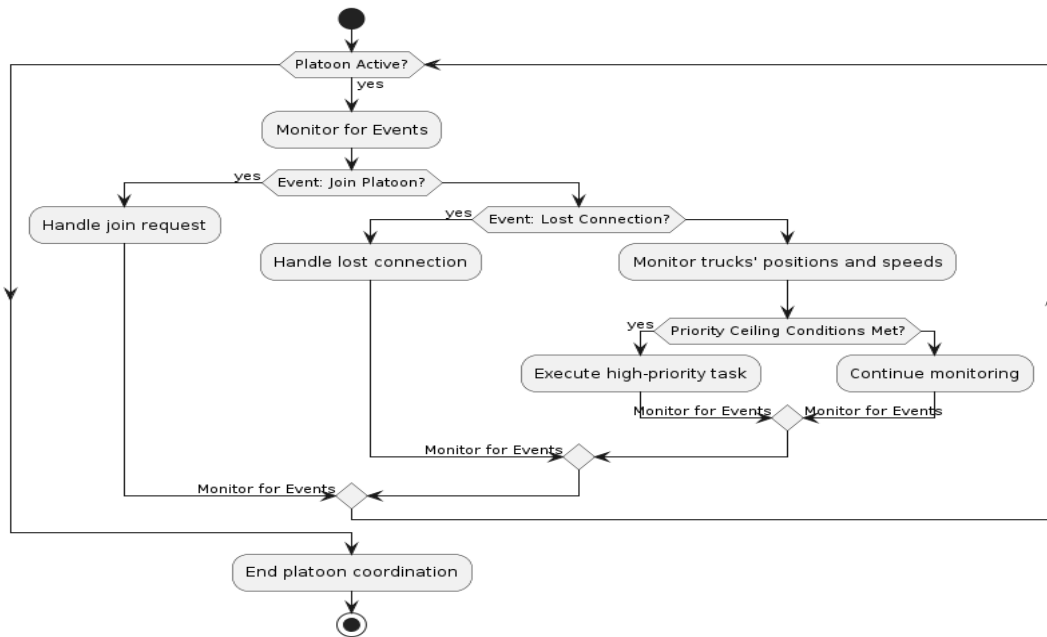
In designing a truck platooning system using the Priority Ceiling Pattern within a Microkernel Architecture, the Priority Ceiling Pattern is crucial for preventing priority inversion and ensuring timely execution of critical tasks. The Microkernel Architecture consists of packages such as Clients (trucks), Core (Microkernel), and Services (extensions). In this setup, each truck (Client) interacts with the Core and Services, where the Core manages fundamental functions like resource management and process communication. The Priority Ceiling Pattern assigns a priority ceiling to each resource, which is the highest priority of any task that may lock the resource. When a truck requests to join or leave a platoon, or when there is a need to adjust speeds and distances, the Core ensures that these requests are handled with the appropriate priority, preventing lower-priority tasks from delaying high-priority ones. This pattern is implemented in the scheduling and resource management mechanisms of the Core, ensuring that critical tasks in the truck platooning system are executed promptly, maintaining system stability and efficiency.

Priority Ceiling

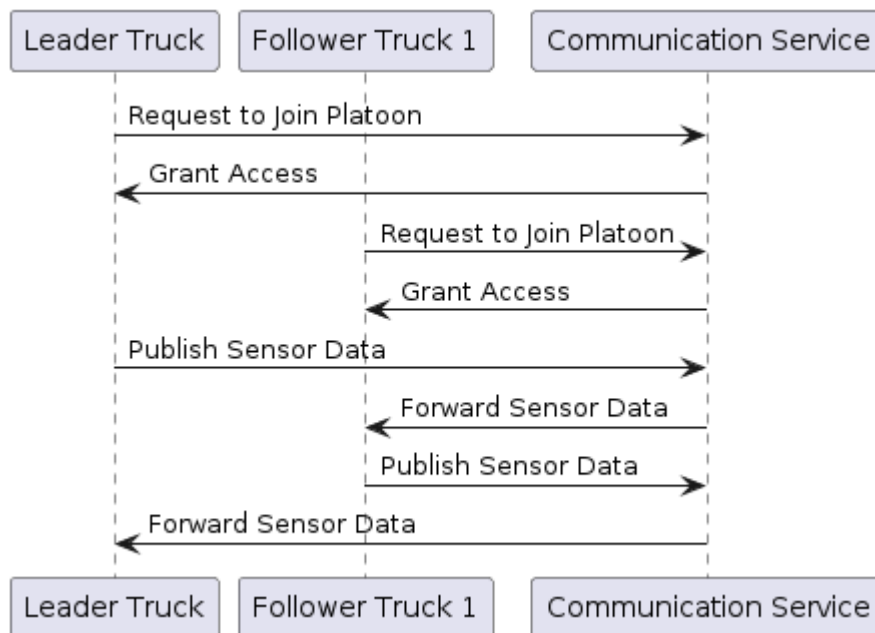
Components Diagram



Activity Diagram



Sequence Diagram Communication Flow



Why does the Priority Ceiling architectural pattern fit best?

The Priority Ceiling pattern is particularly suitable for real-time systems like truck platooning due to the following reasons:

Prevention of Priority Inversion: In real-time systems, priority inversion can occur when a lower-priority task holds a resource needed by a higher-priority task, leading to delays. The Priority Ceiling protocol ensures that a task cannot be preempted by another task of lower priority that also requires the same resource. This prevents priority inversion scenarios, which are critical in ensuring timely and predictable responses in safety-critical applications like truck platooning.

Efficient Resource Management: Truck platooning involves multiple tasks with varying levels of urgency (e.g., handling join requests, monitoring truck positions). By assigning a priority ceiling to resources, the system ensures that higher-priority tasks can proceed without unnecessary delays caused by lower-priority tasks.

Guaranteed Timeliness: Tasks in truck platooning often have strict timing requirements. By using the Priority Ceiling pattern, tasks with critical deadlines are given priority access to resources, ensuring deadlines are met and system responsiveness is maintained.

Why are other alternatives not as good as the Priority Ceiling pattern?

Other resource management patterns, such as Mutexes (Mutual Exclusion) and Semaphores, do not offer the same level of protection against priority inversion:

Mutexes: While Mutexes provide mutual exclusion, they do not prevent priority inversion. If a lower-priority task holds a Mutex and a higher-priority task attempts to acquire it, the higher-priority task may be delayed indefinitely if the lower-priority task does not release the Mutex promptly.

Semaphores: Semaphores can control access to resources using counters, but they do not address priority inversion directly. Higher-priority tasks may still be delayed by lower-priority tasks holding the Semaphore.

Mapping of roles of the Priority Ceiling pattern to the truck platooning system:

In the context of truck platooning, the Priority Ceiling pattern can be mapped as follows:

Resources: Each critical resource (e.g., communication channels, sensor data, control signals) is assigned a priority ceiling.

Tasks: Various tasks within the platooning system (e.g., handling join/leave requests, monitoring truck positions, adjusting speeds) are assigned priorities based on their urgency and criticality.

Protocol: Implementing the Priority Ceiling protocol ensures that tasks holding a resource cannot be preempted by lower-priority tasks that also require the same resource. This guarantees timely execution of critical operations.

Functionality of each component in the architecture:

Here's a brief description of how each component in the truck platooning system functions within the Priority Ceiling architectural pattern:

Communication Module:

Function: Handles communication between trucks in the platoon.

Priority Ceiling Role: Manages access to communication channels ensuring higher-priority messages are transmitted without delay.

Vehicle Control Unit (VCU):

Function: Controls vehicle operations based on platooning commands and sensor inputs.

Priority Ceiling Role: Ensures critical commands (e.g., speed adjustments) are executed promptly without interruption from lower-priority control tasks.

Sensors and Actuators:

Function: Collect sensor data (e.g., speed, distance) and execute control commands (e.g., braking, acceleration).

Priority Ceiling Role: Access to sensor data and actuator controls is managed to prioritize critical operations such as emergency braking or maintaining safe platooning distances.

Platooning Manager:

Function: Coordinates overall platoon operations, manages join/leave requests, and monitors platoon status.

Priority Ceiling Role: Handles critical platoon management tasks, ensuring uninterrupted monitoring and response to platoon events.

Safety Systems:

Function: Implements safety protocols and emergency procedures.

Priority Ceiling Role: Priority access to safety-critical functions such as collision avoidance and emergency braking to prevent accidents within the platoon.

By applying the Priority Ceiling pattern, the truck platooning system can efficiently manage resource access and task scheduling, ensuring reliable and timely operations critical for safe and effective platooning on the road.